CS112 - Fall 2022
Lab18
Instructor: Paul Haskell

INTRODUCTION

Let's write a program that takes advantage of polymorphism!

Polymorphism

This next assignment is kind of silly but it is a classic in the history of Polymorphism homework. You're not the first ones to do this.

You know the old children's song "Old McDonald Had a Farm"? It is used to entertain small children and teach them the names of some animals and the sounds they make. A few verses are:

> Old McDonald Had a Farm, EIEIO,
> And on his farm he had a cow, EIEIO,
> With a moo, moo here, a moo, moo there…
>
> Old McDonald Had a Farm, EIEIO,
> And on his farm he had a dog, EIEIO,
> With a bark, bark here, a bark, bark there…
>
> Old McDonald Had a Farm, EIEIO,
> And on his farm he had a goose, EIEIO,
> With a honk, honk here, a honk, honk there…

You can probably predict where this is going.

I have given you the beginnings of a program called **McDonald.java** . The program contains an abstract class `Animal` . Your job is to finish the program.

Please derive five animal classes of your choosing from `class Animal`. Please pick animals that make some kind of sound (e.g. not earthworms or clams).   Give proper definitions for `String sound()` for each animal. And overwrite the `toString()` method to return the name of your animal.

Ok, now your **McDonald.java** can use all your animal classes to make the song. But let's generate the song at random! If you use the following code:

```
int randomIntFromZeroThroughFour = (int) (5*Math.random());
```

you will get a random integer in the rage [0, 4] every time this statement is executed.

Please use something like this statement to print out exactly 10 verses of Old McDonald, at random, using your five animals. You can follow the template above exactly for your output, but include your personal choice of animal names and sounds.

- Please put a blank line between verses

Reminder

Put all your files in **Lab18** and push to GitHub before the deadline. This assignment must be turned in before 11:59pm on Friday November 4th.

Conclusion

This program gives you some experience making derived classes and using polymorphism. Although the **McDonald.java** program seems a bit silly, the way it works is quite important: your software that prints out the song should be independent of the specific Animal definitions. It should only use and know about the base `class Animal`. You should be able to come back later and add five more animal types, and the song-printing part of the program should remain unchanged. That obviously seems quite useful--but I will try to resist making that next week's homework!

Grading Rubric

McDonald.java is worth 10 points: output should be random, should include 10 verses every time, should include 5 different animals in random order every time. Software design and quality is worth 5 points. In particular, the McDonald song-printing code should be independent of the specific types of Animals used; it should only use the `Animal` base class.